

What is claimed is:

1 1: A method comprising:
2 associating a virtual machine with a processor utilizing a virtual machine
3 identifier;
4 receiving an interrupt;
5 determining if the interrupt is associated with a virtual machine identifier that is
6 associated with one or more processors; and
7 if so, routing the interrupt to the matching processor(s).

1 2: The method of claim 1, wherein determining if the interrupt is associated with a
2 virtual machine identifier that is associated with one or more processors includes:
3 determining if the interrupt is associated with a virtual machine identifier; and
4 if so, determining if the virtual machine identifier is associated with one or more
5 processors.

1 3: The method of claim 2, further comprising:
2 if the interrupt is not associated with a virtual machine identifier,
3 utilizing a virtual control block to steer the interrupt to the appropriate processor.

1 4: The method of claim 1, further comprising:
2 if the interrupt is associated with a virtual machine identifier that is currently not

3 associated with one or more processors,
4 targeting a virtual machine not currently running on any processor; and
5 routing the interrupt to a processor running in shared mode and reported as
6 executing the lowest priority task.

1 5: The method of claim 4, wherein associating a virtual machine with a processor
2 utilizing a virtual machine identifier includes:
3 receiving a communication from the processor that includes a virtual machine
4 identifier; and
5 storing a processor identifier and the virtual machine identifier in a participant
6 table.

1 6: The method of claim 5, wherein receiving a communication from the processor
2 includes utilizing, at least in part, the VM-ID information to optimize system resources
3 and parameters.

1 7: The method of claim 5, wherein receiving a communication from the processor that
2 includes a virtual machine identifier occurs when the processor either initiates or resumes
3 execution of a virtual machine indicated by the virtual machine identifier.

1 8: The method of claim 7, wherein a virtual control block (VCB)
2 specifies a virtual machine identifier for each virtual machine, and
3 stores the virtual machine identifier in an executing processor utilizing a virtual
4 machine control block.

1 9: The method of claim 5, wherein associating a virtual machine with a processor
2 utilizing a virtual machine identifier includes:
3 receiving a communication from the processor that includes information that
4 denotes whether or not the processor is running in shared or dedicated mode; and
5 storing the processor mode information in a participant table.

1 10: The method of claim 1, wherein routing the interrupt to the matching processor(s)
2 includes:
3 if the virtual machine identifier is associated with multiple processors, routing the
4 interrupt to the associated processor with the lowest task priority.

1 11: The method of claim 1, further including:
2 associating an interrupt generating device that is exclusively assigned to a virtual
3 machine with the virtual machine's identifier.

1 12: The method of claim 11, wherein associating an interrupt generating device includes
2 storing the virtual machine identifier in a memory element within the device so that any
3 interrupts generated by the device may include the virtual machine identifier.

1 13: The method of claim 11, wherein associating an interrupt generating device includes:
2 utilizing an interrupt controller, having interrupt input lines, to route all interrupts
3 from the interrupt generating device, and
4 associating a virtual machine identifier with an input line of the interrupt
5 controller; and
6 wherein the interrupt controller assumes that all interrupts incoming on the associated
7 interrupt input line are associated with the virtual machine identifier.

1 14: An article comprising:
2 a machine accessible medium having a plurality of machine accessible instructions,
3 wherein when the instructions are executed, the instructions provide for:
4 associating a virtual machine with a processor utilizing a virtual machine
5 identifier;
6 receiving an interrupt;
7 determining if the interrupt is associated with a virtual machine identifier that is
8 associated with one or more processors; and
9 if so, routing the interrupt to the matching processor(s).

1 15: The article of claim 14, wherein the instructions providing for determining if the
2 interrupt is associated with a virtual machine identifier that is associated with one or more
3 processors includes instructions providing for:
4 determining if the interrupt is associated with a virtual machine identifier; and
5 if so, determining if the virtual machine identifier is associated with one or more
6 processors.

1 16: The article of claim 15, further comprising instructions providing for:
2 if the interrupt is not associated with a virtual machine identifier,
3 utilizing a virtual control block to steer the interrupt to the appropriate processor.

1 17: The article of claim 14, further comprising instructions providing for:
2 if the interrupt is not associated with a virtual machine identifier that is currently
3 associated with one or more processors,
4 targeting a virtual machine not currently running on any processor; and
5 routing the interrupt to a processor running in shared mode and reported as
6 executing the lowest priority task.

1 18: The article of claim 17, wherein the instructions providing for associating a virtual
2 machine with a processor utilizing a virtual machine identifier includes instructions
3 providing for:

4 receiving a communication from the processor that includes a virtual machine
5 identifier; and

6 storing a processor identifier and the virtual machine identifier in a participant
7 table.

1 19: The article of claim 18, wherein the instructions providing for receiving a
2 communication from the processor includes instructions providing for utilizing, at least in
3 part, the VM-ID information to optimize system resources and parameters.

1 20: The article of claim 18, wherein the instructions providing for receiving a
2 communication from the processor that includes a virtual machine identifier occurs when
3 the processor either initiates or resumes execution of a virtual machine indicated by the
4 virtual machine identifier.

1 21: The article of claim 20, further including instructions providing for a virtual control
2 block (VCB)
3 specifying a virtual machine identifier for each virtual machine, and

4 storing the virtual machine identifier in an executing processor utilizing a virtual
5 machine control block.

1 22: The article of claim 18, wherein the instructions providing for associating a virtual
2 machine with a processor utilizing a virtual machine identifier includes instructions
3 providing for:

4 receiving a communication from the processor that includes information that
5 denotes whether or not the processor is running in shared or dedicated mode; and
6 storing the processor mode information in a participant table.

1 23: The article of claim 14, wherein the instructions providing for routing the interrupt to
2 the matching processor(s) includes instructions providing for:

3 if the virtual machine identifier is associated with multiple processors, routing the
4 interrupt to the associated processor with the lowest task priority.

1 24: The article of claim 14, further including instructions providing for:

2 associating an interrupt generating device that is exclusively assigned to a virtual
3 machine with the virtual machine's identifier.

1 25: The article of claim 24, wherein the instructions providing for associating an
2 interrupt generating device includes instructions providing for storing the virtual machine
3 identifier in a memory element within the device so that any interrupts generated by the
4 device may include the virtual machine identifier.

1 26: The article of claim 14, wherein the instructions providing for associating an
2 interrupt generating device includes instructions providing for:
3 utilizing an interrupt controller, having interrupt input lines, to route all interrupts
4 from the interrupt generating device, and
5 associating a virtual machine identifier with an input line of the interrupt
6 controller; and
7 wherein the interrupt controller assumes that all interrupts incoming on the associated
8 interrupt input line are associated with the virtual machine identifier.

1 27: An apparatus comprising:
2 a plurality of processors to execute a plurality of virtual machines having virtual
3 machine identifiers; and
4 an integrated circuit to steer interrupts to the processor utilizing, at least in part,
5 the virtual machine interrupts.

1 28: The apparatus of claim 27, wherein each processor is capable of:

2 communicating, to the integrated circuit, the virtual machine identifier of the
3 virtual machine that the processor is currently executing.

1 29: The apparatus of claim 28, wherein the integrated circuit is capable of associating
2 each processor with a virtual machine identifier, and the associations are stored in a
3 participant table.

1 30: The apparatus of claim 29, wherein the integrated circuit is further capable of
2 determining whether each processor is running in shared or dedicated mode.

1 31: The apparatus of claim 27, wherein the integrated circuit is capable of:
2 receiving an interrupt;
3 determining if the interrupt is associated with a virtual machine identifier that is
4 associated with one or more processors; and
5 if so, routing the interrupt to the matching processor(s).

1 32: The apparatus of claim 31, wherein the integrated circuit is capable of utilizing, at
2 least in part, the VM-ID information to optimize system resources and parameters.

1 33: The apparatus of claim 31, wherein the integrated circuit is further capable of:
2 if the interrupt is not associated with a virtual machine identifier,
3 utilizing a virtual control block to steer the interrupt to the appropriate processor.

1 34: The apparatus of claim 33, further including a virtual control block (VCB) that is
2 capable of:
3 specifying a virtual machine identifier for each virtual machine, and
4 storing the virtual machine identifier in an executing processor utilizing a virtual
5 machine control block.

1 35: The apparatus of claim 33, wherein the integrated circuit is capable of:
2 if the interrupt is associated with a virtual machine identifier that is not currently
3 associated with one or more processors,
4 targeting a virtual machine not currently running on any processor; and
5 routing the interrupt to a processor running in shared mode reported as executing
6 the lowest priority task.

1 36: The apparatus of claim 31, wherein the integrated circuit is capable of:
2 if the virtual machine identifier is associated with multiple processors, routing the
3 interrupt to the associated processor with the lowest task priority.

1 37: The apparatus of claim 27, wherein the each processor includes a memory element to
2 store the virtual machine identifier of the virtual machine that is currently being executed.

1 38: The apparatus of claim 28, wherein each processor is capable of:
2 communicating, to the integrated circuit, the virtual machine identifier of the
3 virtual machine that the processor is currently executing when the processor either
4 initiates or resumes execution of the virtual machine.

1 39: The apparatus of claim 28, wherein the integrated circuit is capable of associating an
2 interrupt generating device by:
3 utilizing an interrupt controller, having interrupt input lines, to route all interrupts
4 from the interrupt generating device, and
5 associating a virtual machine identifier with an input line of the interrupt
6 controller; and
7 wherein the interrupt controller assumes that all interrupts incoming on the associated
8 interrupt input line are associated with the virtual machine identifier.

1 40: A system comprising:
2 a plurality of processors to execute a plurality of virtual machines having virtual

3 machine identifiers;
4 at least one interrupt generating device to transmit an interrupt having a virtual
5 machine identifier; and
6 an integrated circuit to steer interrupts to the processor utilizing, at least in part,
7 the virtual machine interrupts.

1 41: The system of claim 40, wherein each processor is capable of:
2 communicating, to the integrated circuit, the virtual machine identifier of the
3 virtual machine that the processor is currently executing.

1 42: The system of claim 41, wherein the integrated circuit is capable of associating each
2 processor with a virtual machine identifier, and the associations are stored in a participant
3 table.

1 43: The system of claim 40, wherein the integrated circuit is further capable of
2 determining whether each processor is running in shared or dedicated mode.

1 44: The system of claim 40, wherein the integrated circuit is capable of:
2 receiving an interrupt;
3 determining if the interrupt is associated with a virtual machine identifier that is

4 associated with one or more processors; and
5 if so, routing the interrupt to the matching processor(s).

1 45: The system of claim 44, wherein the integrated circuit is capable of utilizing, at least
2 in part, the VM-ID information to optimize system resources and parameters.

1 46: The system of claim 44, wherein the integrated circuit is further capable of:
2 if the interrupt is not associated with a virtual machine identifier,
3 utilizing a virtual control block to steer the interrupt to the appropriate processor.

1 47: The system of claim 46, further including a virtual control block (VCB) that is
2 capable of:
3 specifying a virtual machine identifier for each virtual machine, and
4 storing the virtual machine identifier in an executing processor utilizing a virtual
5 machine control block.

1 48: The system of claim 46, wherein the integrated circuit is capable of:
2 if the interrupt is associated with a virtual machine identifier that is not currently
3 associated with one or more processors,
4 targeting a virtual machine not currently running on any processor; and

5 routing the interrupt to a processor running in shared mode reported as executing
6 the lowest priority task.

1 49: The system of claim 44, wherein the integrated circuit is capable of:
2 if the virtual machine identifier is associated with multiple processors, routing the
3 interrupt to the associated processor with the lowest task priority.

1 50: The system of claim 40, wherein the each processor includes a memory element to
2 store the virtual machine identifier of the virtual machine that is currently being executed.

1 51: The system of claim 40, wherein the interrupt generating device is capable of:
2 being exclusively used by a single virtual machine;
3 receiving the virtual machine identifier of the virtual machine that has exclusive
4 use of the device; and
5 the interrupt generating device includes a memory element to store the virtual machine
6 identifier.

1 52: The system of claim 41, wherein the integrated circuit is capable of associating an
2 interrupt generating device by:
3 utilizing an interrupt controller, having interrupt input lines, to route all interrupts

4 from the interrupt generating device, and
5 associating a virtual machine identifier with an input line of the interrupt
6 controller; and
7 wherein the interrupt controller assumes that all interrupts incoming on the associated
8 interrupt input line are associated with the virtual machine identifier.